Analog.7020
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Boutaud		
Serial No.:	10/715,629	**Group No:**	2181
Filed:	11/17/2003	**Examiner:**	V. Lai
For:	DIGITAL SIGNAL PROCESSOR ARCHITECTURE WITH OPTIMIZED MEMORY ACCESS FOR CODE DISCONTINUITY		

Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF (PATENT APPLICATION--37 CFR 192)

1. Transmitted herewith is the APPEAL BRIEF in this application with respect to the Notice of Appeal filed on 11/17/2006.

NOTE: "The appellant shall, within 2 months from the date of the notice of appeal under 1.191 in an application, reissue application, or patent under reexamination, or within the time allowed for response to the action appealed from, if such time is later, file a brief *in triplicate*." 37 CFR 1.192(a) [emphasis added]

2. STATUS OF APPLICANT

This application is on behalf of

X other than a small entity

_ small entity
verified statement:

_ attached

_ already filed

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 CFR 1.17(f) the fee for filing the Appeal Brief is:

_ small entity $250.00

X other than a small entity $500.00

Appeal Brief fee due $ 500.00

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Patent and Trademark Office, via EFS-Web on the date shown below.

Date: 1-16-07

Elizabeth M. Ball
(type or print name of person mailing paper)
Elizabeth M Ball
(Signature of person mailing paper)

4. EXTENSION OF TERM

NOTE: The time periods set forth in 37 CFR 1.192(a) are subject to the provision of 1.136 for patent applications. 37 CFR 1.191(d). Also see Notice of November 5, 1985 (1060 O.G. 27).

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply.

(complete (a) or (b) as applicable)

(a) _ Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-d)) for the total number of months checked below:

	Extension (months)	Fee for other than small entity	Fee for small entity
_	one month	$110.00	$55.00
_	two months	$390.00	$195.00
_	three months	$930.00	$465.00
_	four months	$1,470.00	$735.00
			Fee $

If an additional extension of time is required please consider this a petition therefor.

(check and complete the next item, if applicable)

_ An extension for ______ months has already been secured and the fee paid therefor of $ is deducted from the total fee due for the total months of extension now requested.

Extension fee due with this request $

or

(b) X Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE

The total fee due is:

Appeal brief fee $ 500.00
Extension fee (if any) $

TOTAL FEE DUE $ 500.00

6. FEE PAYMENT

_ Attached is a check in the sum of $______.

X Charge Account No. 19-0079 the sum of $500.00.

A duplicate of this transmittal is attached.

7. FEE DEFICIENCY

NOTE: If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum, six month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays encountered in returning the papers to the PTO Finance Branch in order to apply these charges prior to action on the cases. Authorization to charge the deposit account for any fee deficiency should be checked. See the Notice of April 7, 1986, 1065 O.G. 31-33.

X If any additional extension and/or fee is required, this is a request therefor and to charge Account No. 19-0079 .

Respectfully submitted,

Matthew E. Connors
Registration No. 33,298
Gauthier & Connors LLP
225 Franklin Street, Suite 2300
Boston, Massachusetts 02110
Telephone: (617) 426-9180, Ext. 112
E-mail: mconnors@gc-law.com

Patent Application Number: 10/715,629
Attorney Docket Number: Analog.7020

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

On behalf of

Frederic **BOUTAUD**

APPELLANT

Application: **10/715,629** Examiner: **V. Lai**

Filed: **November 17, 2003** Group Art Unit: **2181**

Title: **DIGITAL SIGNAL PROCESSOR ARCHITECTURE WITH OPTIMIZED MEMORY ACCESS FOR CODE DISCONTINUITY**

APPELLANT'S BRIEF ON APPEAL

TABLE OF CONTENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT:	Frederic BOUTAUD	GROUP:	2181
APPLICATION:	10/715,629	EXAMINER:	V. Lai
FILED:	November 17, 2003	CONFIRMATION:	4322

FOR: **DIGITAL SIGNAL PROCESSOR ARCHITECTURE WITH OPTIMIZED MEMORY ACCESS FOR CODE DISCONTINUITY**

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313–1450

Sir:

APPEAL BRIEF FOR APPELLANTS

This Appeal Brief is being submitted in accordance with the Notice of Appeal filed on November 13, 2006 in connection with the above-identified application.

I. REAL PARTY OF INTEREST

The party of real interest to this appeal is the Assignee, Analog Devices, Incorporated.

II. RELATED APPEALS AND INTERFERENCES

The Appellant knows of no other pending appeals or interferences that are related to this instant appeal.

III. STATUS OF CLAIMS

Claims 1-14 have been previously presented in this application. Claims 1-14 are appealed.

IV. STATUS OF AMENDMENTS

The Appellant submitted a Response under 37 C.F.R. 1.116 on October 24, 2006, wherein the Appellant made no amendments to the claims. The Appellant filed a Request for a Pre-Appeal Brief Review on November 13, 2006. The Appellant received a Decision dated January 10, 2007. The Appellant has not filed any other Responses and/or Amendments subsequent to the Final Office Action, dated July 28, 2006.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In accordance with 37 C.F.R. 41.37(2)(c)(v), the following are concise explanations of the subject matter defined in the independent claims involved in this Appeal.

A. Independent Claim 1

Independent claim 1 recites a method for accessing a unified memory in a micro-processing system having a microprocessor, a one level pipeline, and a two-phase clock, such that an instruction is executed in a single instruction cycle (see, for example, Figure 16 and paragraph **[0030]** of the published patent application). The method fetches a program instruction from the unified memory and determines if the fetched program instruction would require three unified memory accesses during a single instruction cycle for proper execution of the fetched program instruction, proper execution of the fetched program instruction being the microprocessor performing operations requested by the fetched program instruction in a single instruction cycle (see, for example, paragraph **[0036]** of the published patent application). The method also accesses the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access (see, for example, paragraphs **[0045]** and **[0046]** of the published patent application) when it is determined that the fetched program instruction requires three unified memory accesses for proper execution of the fetched program instruction and fetches a next program instruction from an instruction register, during the instruction cycle associated with the fetched program instruction from the unified memory, when it is determined that the fetched program instruction requires three unified memory accesses for proper execution of the fetched program instruction (see, for example, Figure 6 and paragraphs **[0057]** through **[0059]** of the published patent application). The method accesses the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction

from the first access of the unified memory (see, for example, Figure 6 and paragraphs **[0057]** through **[0059]** of the published patent application).

B. Independent Claim 6

Independent claim 6 recites a method for accessing a unified memory in a micro-processing system having a microprocessor, a one level pipeline, and a two-phase clock, such that an instruction is executed in a single instruction cycle (see, for example, Figure 16 and paragraph **[0030]** of the published patent application). The method fetches a program instruction, corresponding to a second instruction cycle, from the unified memory during a first instruction cycle and determines if the fetched program instruction corresponding to the second instruction cycle is a conditional program code discontinuity (see, for example, Figure 7 and paragraphs **[0060]** through **[0065]** of the published patent application). The method also accesses the unified memory a first time during the second instruction cycle with a dummy access (see, for example, paragraphs **[0045]** and **[0046]** of the published patent application) when it is determined that the fetched program instruction corresponding to the second instruction cycle is a conditional program code discontinuity (see, for example, Figure 7 and paragraphs **[0060]** through **[0065]** of the published patent application) and accesses the unified memory a second time during the second instruction cycle to read a new instruction when it is determined the fetched program instruction corresponding to the second instruction cycle is a conditional program code discontinuity, thereby delaying instruction access from the unified memory for the second instruction cycle by a half cycle (see, for example, Figure 7 and paragraphs **[0060]** through **[0065]** of the published patent application).

C. Independent Claim 9

Independent claim 9 recites a method for accessing a unified memory in a micro-processing system having a microprocessor, a one level pipeline, and a two-phase clock, such that an instruction is executed in a single instruction cycle (see, for example, Figure 16 and paragraph **[0030]** of the published patent application). The method fetches a program instruction from the unified memory; determines if the fetched program instruction is a loop initiation instruction (see, for example, Figure 11 and paragraphs **[0078]** through **[0082]** of the published patent application); stores a first instruction of the loop in an instruction register when the fetched program instruction is a loop initiation instruction (see, for example, Figure 11 and paragraphs **[0078]** through **[0082]** of the published patent application); and executes the loop. The method also determines if a fetched instruction during the execution of the loop is a last instruction of the loop (see, for example, Figure 11 and paragraphs **[0078]** through **[0082]** of the published patent application); accesses the unified memory a first time, during the instruction cycle associated with the fetched last instruction of the loop, with a dummy access (see, for example, Figure 11 and paragraphs **[0078]** through **[0082]** of the published patent application); fetches the first instruction of the loop from the instruction register, during the instruction cycle associated with the fetched last instruction of the loop (see, for example, Figure 11 and paragraphs **[0078]** through **[0082]** of the published patent application); and accesses the unified memory a second time, during the instruction cycle associated with the fetched last instruction of the loop, with a data access (see, for example, Figure 11 and paragraphs **[0078]** through **[0082]** of the published patent application).

D. Independent Claim 13

Independent claim 13 recites a method for accessing a unified memory in a micro-processing system during execution of a loop instruction. The method accesses a program instruction from the unified memory during a first instruction cycle; determines a type of program instruction; pre-fetches a second instruction from the unified memory; and saves the pre-fetched instruction in a register when it is determined that the type of program instruction is a first instruction of a loop (see, for example, Figure 12 and paragraphs **[0083]** through **[0093]** of the published patent application). The method also fetches an instruction from the register when it is determined that the type of program instruction is a last instruction of a loop; accesses the unified memory with a dummy access during execution of the last instruction of the loop; and accesses the unified memory, a second time, with a data access during execution of the last instruction of the loop (see, for example, Figure 12 and paragraphs **[0083]** through **[0093]** of the published patent application).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Rejection under 35 U.S.C. §102(b) over Morley

The grounds of rejection to be reviewed on appeal are whether claims 1-14 are patentable over Morley (US-A-4,276,594) in accordance with 35 U.S.C. §102(b).

VII. ARGUMENTS

A. Rejection of claims 1-14 under 35 U.S.C. §102(b) over Morley

Claims 1-14 have been rejected under 35 U.S.C. §102(b) as being anticipated by Morley (US-A-4,276,594). This rejection is respectfully traversed.

A. Independent Claim 1

With respect to independent claim 1, the Examiner alleges that Morley discloses fetching a program instruction from the unified memory; determining if the fetched program instruction would require three unified memory accesses during a single instruction cycle; accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access; fetching a next program instruction from an instruction register, during the instruction cycle associated with the fetched program instruction; and accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction, with a data access. From the allegations, the Examiner concludes that the presently claimed invention of independent claim 1 is anticipated by the teachings of Morley.

As respectfully submitted above, independent claim 1 recites a method for accessing a unified memory in a micro-processing system having a microprocessor, a one level pipeline, and a two-phase clock, such that an instruction is executed in a single instruction cycle. The method fetches a program instruction from the unified memory and determines if the fetched program instruction would require three unified memory accesses during a single instruction cycle for proper execution of the fetched program instruction, proper execution of the fetched program instruction being the microprocessor performing operations requested by the fetched program instruction in a single instruction cycle. The method also accesses the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access when it is determined that the fetched program instruction requires three unified memory accesses for proper execution of the fetched program instruction and

fetches a next program instruction from an instruction register, during the instruction cycle associated with the fetched program instruction from the unified memory, when it is determined that the fetched program instruction requires three unified memory accesses for proper execution of the fetched program instruction. The method accesses the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory.

As noted above, the Examiner alleges that Morley discloses determining if the fetched program instruction would require three unified memory accesses during a single instruction cycle and points to column 58, lines 17-20, of Morley to support the allegation. Column 58, lines 17-20, states:

> The 6800 accesses this RAM during the PHΦ of the MIO cycle. During serial I/O, this time is used to fetch/store characters and update pointers. For parallel I/O, this is used to fetch parameters for the PIA.

Contrary to the Examiner' position, this passage of Morley fails to discuss or disclose any determination with respect to the number of unified memory accesses that would be required during a single instruction cycle. Thus, this passage of Morley cannot provide any basis for establishing anticipation with respect to determining if the fetched program instruction would require three unified memory accesses during a single instruction cycle for proper execution of the fetched program instruction, proper execution of the fetched program instruction being the microprocessor performing operations requested by the fetched program instruction in a single instruction cycle, as set forth by independent claim 1.

At column 58, lines 17-20, Morley teaches that the RAM includes a parallel I/O port and a serial I/O port. Moreover, Morley teaches that the RAM is access through the Serial I/O and the Parallel I/O to fetch data from the memory. Notwithstanding the

use of both the Serial I/O and the Parallel I/O to fetch data from the memory, Morley fails to teach determining if the fetched program instruction would require three unified memory accesses during a single instruction cycle for proper execution of the fetched program instruction since only two fetches are discussed.

Therefore, Morley is void of any teachings directed to determining if the fetched program instruction would require three unified memory accesses during a single instruction cycle for proper execution of the fetched program instruction, proper execution of the fetched program instruction being the microprocessor performing operations requested by the fetched program instruction in a single instruction cycle, as set forth by independent claim 1.

In response to this argument, the Examiner states that it is the Examiner's interpretation of the teachings of Morley that provides the basis for the Examiner's allegation that "the number of unified access [sic] **can be** determined and three unified accesses **is** [sic] **possible.**" [Emphasis added.] The Examiner has failed to provide any technical or factual basis to support the Examiner's allegation that Morley teaches determining if the fetched program instruction would require three unified memory accesses during a single instruction cycle for proper execution of the fetched program instruction. Anticipation is based what is taught by the reference, not speculation as to what the prior art may or may not be able to do.

In other words, absent any specific passages of in Morley that teach making a request to determine the number of unified memory accesses that would be required during a single instruction cycle and follow-up determination, the Examiner's assertions are mere conjecture of what Morley could possibly teach as the Examiner has failed to provide any specific evidence to the contrary.

The Examiner also alleges that Morley discloses accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access.

The originally filed specification sets forth that "the cycle steal occurs at an instruction boundary by holding the next instruction from the program count and inserting a special "dummy" instruction into the pipeline," and " this dummy instruction does not change the program count nor starts a program instruction fetch."

On the other hand, contrary to the Examiner' position, Morley, at column 57 lines 9-25, teaches that although the RAM has three ports, only one port can be accessed at any time. Moreover, Morley teaches that four cycles are needed to transfer the 32-bit data. In other words, Morley fails to discuss or disclose any type of dummy access of the unified memory.

Thus, Morley fails to provide any basis for a finding of anticipation to the limitation corresponding to accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access of independent claim 1.

Lastly, as noted above, the Examiner alleges that Morley discloses accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction, with a data access and points to Figure 28 to support the allegation.

With respect to Figure 28, the read operation illustrates that during the PHΦ of the MIO cycle, the data from the memory is accessed and placed on the bus; during the PH1 of the MIO cycle, the data sits on the bus; and during the PH2 of the MIO cycle, the data on the bus is uploaded or read. Thus, the memory is only access once per MIO cycle and fails to anticipate accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction, with a data access, as set forth by independent claim 1.

In summary, with respect to independent claim 1, Morley fails to teach (1) determining if the fetched program instruction would require three unified memory accesses during a single instruction cycle for proper execution of the fetched program instruction; (2) accessing the unified memory a first time, during the instruction cycle

associated with the fetched program instruction, with a dummy access; and/or (3) accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction, with a data access.

Therefore, Morley fails to anticipate the claimed invention of independent claim 1.

B. Dependent Claim 2

With respect to dependent claim 2, the Examiner points to Figure 28 and alleges that Morley discloses that a data access is a read data access.

Dependent claim 2 further limits claim 1 such that the claimed method accesses the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a read data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory.

In contrast, Figure 28 of Morley merely illustrates the existence of read data accesses. Figure 28 of Morley fails to disclose or illustrate accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a read data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory, as established by dependent claim 2.

Therefore, Morley fails to anticipate the claimed invention of dependent claim 2.

C. Dependent Claim 3

With respect to dependent claim 3, the Examiner points to Figure 28 and alleges that Morley discloses that a data access is a write data access.

Dependent claim 3 further limits claim 1 such that the claimed method accesses the unified memory a second time, during the instruction cycle associated with the

fetched program instruction from the first access of the unified memory, with a write data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory.

In contrast, Figure 28 of Morley merely illustrates the existence of write data accesses. Figure 28 of Morley fails to disclose or illustrate accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a write data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory, as established by dependent claim 3.

Therefore, Morley fails to anticipate the claimed invention of dependent claim 3.

D. Dependent Claim 4

With respect to dependent claim 4, the Examiner points to column 41, line 64 to column 65, line 1 and alleges that Morley discloses that the fetched program instruction from the first access of the unified memory is a last instruction of a loop.

Dependent claim 4 further limits claim 1 such that the claimed method accesses the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access when it is determined that the fetched program instruction requires three unified memory accesses for proper execution of the fetched program instruction to retrieve a last instruction of a loop.

In contrast, column 41, line 64 to column 65, line 1 of Morley merely teaches the capability of handling a loop. Column 41, line 64 to column 65, line 1 of Morley fails to disclose accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access when it is determined that the fetched program instruction requires three unified memory

accesses for proper execution of the fetched program instruction to retrieve a last instruction of a loop, as established by dependent claim 4.

Therefore, Morley fails to anticipate the claimed invention of dependent claim 4.

E. Independent Claim 6

With respect to independent claim 6, the Examiner alleges that Morley discloses fetching a program instruction from the unified memory during a first instruction cycle; determining if the fetched program instruction for a second instruction cycle is a conditional program code discontinuity; accessing the unified memory a first time during the second instruction cycle with a dummy access when it is determined that the program instruction accessed for a second instruction cycle is a conditional program code discontinuity; and accessing the unified memory a second time during the second instruction cycle to read a new instruction when it is determined the program instruction accessed for a second instruction cycle is a conditional program code discontinuity, thereby delaying the instruction access from the unified memory for the second instruction cycle by a half cycle. From the allegations, the Examiner concludes that the presently claimed invention of independent claim 6 is anticipated by the teachings of Morley.

As noted above, the Examiner alleges that Morley discloses accessing the unified memory a first time during the second instruction cycle with a dummy access when it is determined that the program instruction accessed for a second instruction cycle is a conditional program code discontinuity and points to column 58, lines 17-20, of Morley to support the allegation. Column 58, lines 17-20, states:

> The 6800 accesses this RAM during the PHΦ of the MIO cycle. During serial I/O, this time is used to fetch/store characters and update pointers. For parallel I/O, this is used to fetch parameters for the PIA.

Contrary to the Examiner' position, this passage of Morley fails to discuss or disclose accessing the unified memory a first time during the second instruction cycle with a dummy access when it is determined that the program instruction accessed for a second instruction cycle is a conditional program code discontinuity, as set forth by independent claim 6.

Column 58, lines 17-20, of Morley merely teaches that the RAM may include a parallel I/O port. Column 58, lines 17-20, of Morley is void of any teachings directed to accessing the unified memory a first time during the second instruction cycle with a dummy access when it is determined that the program instruction accessed for a second instruction cycle is a conditional program code discontinuity, as set forth by independent claim 6.

The Examiner also alleges that Morley discloses accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access.

The originally filed specification sets forth that "the cycle steal occurs at an instruction boundary by holding the next instruction from the program count and inserting a special "dummy" instruction into the pipeline," and " this dummy instruction does not change the program count nor starts a program instruction fetch."

On the other hand, contrary to the Examiner' position, Morley, at column 57 lines 9-25, teaches that although the RAM has three ports, only one port can be accessed at any time. Moreover, Morley teaches that four cycles are needed to transfer the 32-bit data. In other words, Morley fails to discuss or disclose any type of dummy access of the unified memory.

Thus, Morley fails to provide any basis for a finding of anticipation to the limitation corresponding to accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access.

Lastly, as noted above, the Examiner alleges that Morley discloses accessing the unified memory a second time during the second instruction cycle to read a new

instruction when it is determined the program instruction accessed for a second instruction cycle is a conditional program code discontinuity, thereby delaying the instruction access from the unified memory for the second instruction cycle by a half cycle and points to Figure 28 to support the allegation.

With respect to Figure 28, the read operation illustrates that during the PHΦ of the MIO cycle, the data from the memory is accessed and place on the bus; during the PH1 of the MIO cycle, the data sits on the bus; and during the PH2 of the MIO cycle, the data on the bus is uploaded or read. Thus, the memory is only access once per MIO cycle and fails to anticipate accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction, with a data access, as set forth by independent claim 6.

In summary, with respect to independent claim 6, Morley fails to teach (1) accessing the unified memory a first time during the second instruction cycle with a dummy access when it is determined that the program instruction accessed for a second instruction cycle is a conditional program code discontinuity; and/or (2) accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction, with a data access.

F. Independent Claim 9

With respect to independent claim 9, the Examiner alleges that Morley discloses fetching a program instruction from the unified memory; determining if the fetched program instruction is a loop initiation instruction; storing a first instruction of the loop in an instruction register when the fetched program instruction is a loop initiation instruction; executing the loop; determining if a fetched instruction during the execution of the loop is a last instruction of the loop; accessing the unified memory a first time, during the instruction cycle associated with the fetched last instruction of loop, with a dummy access; fetching the first instruction of the loop from the instruction register, during the instruction cycle associated with the fetched last instruction of loop; and accessing the unified memory a second time, during the instruction cycle associated

with the fetched last instruction of loop, with a data access. From the allegations, the Examiner concludes that the presently claimed invention of independent claim 9 is anticipated by the teachings of Morley.

As noted above, the Examiner alleges that Morley discloses accessing the unified memory a first time, during the instruction cycle associated with the fetched last instruction of loop, with a dummy access and points to column 58, lines 17-20, of Morley to support the allegation. Column 58, lines 17-20, states:

> The 6800 accesses this RAM during the PHΦ of the MIO cycle. During serial I/O, this time is used to fetch/store characters and update pointers. For parallel I/O, this is used to fetch parameters for the PIA.

Contrary to the Examiner' position, this passage of Morley fails to discuss or disclose accessing the unified memory a first time, during the instruction cycle associated with the fetched last instruction of loop, with a dummy access, as set forth by independent claim 9.

Column 58, lines 17-20, of Morley merely teaches that the RAM may include a parallel I/O port. Column 58, lines 17-20, of Morley is void of any teachings directed to accessing the unified memory a first time, during the instruction cycle associated with the fetched last instruction of loop, with a dummy access, as set forth by independent claim 9.

The Examiner also alleges that Morley discloses accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access.

The originally filed specification sets forth that "the cycle steal occurs at an instruction boundary by holding the next instruction from the program count and inserting a special "dummy" instruction into the pipeline," and " this dummy instruction does not change the program count nor starts a program instruction fetch."

On the other hand, contrary to the Examiner' position, Morley, at column 57 lines 9-25, teaches that although the RAM has three ports, only one port can be accessed at

any time. Moreover, Morley teaches that four cycles are needed to transfer the 32-bit data. In other words, Morley fails to discuss or disclose any type of dummy access of the unified memory.

Thus, Morley fails to provide any basis for a finding of anticipation to the limitation corresponding to accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access.

Moreover, as noted above, the Examiner alleges that Morley discloses accessing the unified memory a second time, during the instruction cycle associated with the fetched last instruction of loop, with a data access and points to Figure 28 to support the allegation.

With respect to Figure 28, the read operation illustrates that during the PHΦ of the MIO cycle, the data from the memory is accessed and place on the bus; during the PH1 of the MIO cycle, the data sits on the bus; and during the PH2 of the MIO cycle, the data on the bus is uploaded or read. Thus, the memory is only access once per MIO cycle and fails to anticipate accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction, with a data access, as set forth by independent claim 9.

In summary, with respect to independent claim 9, Morley fails to teach (1) accessing the unified memory a first time, during the instruction cycle associated with the fetched last instruction of loop, with a dummy access; and/or (2) accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction, with a data access.

G. Dependent Claim 10

With respect to dependent claim 10, the Examiner points to Figure 28 and alleges that Morley discloses that a data access is a read data access.

Dependent claim 10 further limits claim 9 such that the claimed method accesses the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a read data

access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory.

In contrast, Figure 28 of Morley merely illustrates the existence of read data accesses. Figure 28 of Morley fails to disclose or illustrate accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a read data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory, as established by dependent claim 10.

Therefore, Morley fails to anticipate the claimed invention of dependent claim 10.

H. Dependent Claim 11

With respect to dependent claim 11, the Examiner points to Figure 28 and alleges that Morley discloses that a data access is a write data access.

Dependent claim 11 further limits claim 9 such that the claimed method accesses the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a write data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory.

In contrast, Figure 28 of Morley merely illustrates the existence of write data accesses. Figure 28 of Morley fails to disclose or illustrate accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a write data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched

program instruction from the first access of the unified memory, as established by dependent claim 11.

Therefore, Morley fails to anticipate the claimed invention of dependent claim 11.

I. Independent Claim 13

With respect to independent claim 13, the Examiner alleges that Morley discloses accessing a program instruction from the unified memory during a first instruction cycle; determining a type of program instruction; pre-fetching a next instruction from the unified memory; saving the pre-fetched instruction in a register when it is determined that the type of program instruction is a first instruction of a loop; fetching a next instruction from the register when it is determined that the type of program instruction is a last instruction of a loop; accessing the unified memory with a dummy access during execution of the last instruction of the loop; and accessing the unified memory, a second time, with a data access during execution of the last instruction of the loop. From the allegations, the Examiner concludes that the presently claimed invention of independent claim 13 is anticipated by the teachings of Morley.

As noted above, the Examiner alleges that Morley discloses accessing the unified memory with a dummy access during execution of the last instruction of the loop and points to column 58, lines 17-20, of Morley to support the allegation. Column 58, lines 17-20, states:

> The 6800 accesses this RAM during the PHΦ of the MIO cycle. During serial I/O, this time is used to fetch/store characters and update pointers. For parallel I/O, this is used to fetch parameters for the PIA.

Contrary to the Examiner' position, this passage of Morley fails to discuss or disclose accessing the unified memory with a dummy access during execution of the last instruction of the loop as set forth by independent claim 13.

Column 58, lines 17-20, of Morley merely teaches that the RAM may include a parallel I/O port. Column 58, lines 17-20, of Morley is void of any teachings directed to

accessing the unified memory with a dummy access during execution of the last instruction of the loop, as set forth by independent claim 13.

The Examiner also alleges that Morley discloses accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access.

The originally filed specification sets forth that "the cycle steal occurs at an instruction boundary by holding the next instruction from the program count and inserting a special "dummy" instruction into the pipeline," and " this dummy instruction does not change the program count nor starts a program instruction fetch."

On the other hand, contrary to the Examiner' position, Morley, at column 57 lines 9-25, teaches that although the RAM has three ports, only one port can be accessed at any time. Moreover, Morley teaches that four cycles are needed to transfer the 32-bit data. In other words, Morley fails to discuss or disclose any type of dummy access of the unified memory.

Thus, Morley fails to provide any basis for a finding of anticipation to the limitation corresponding to accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access.

Moreover, as noted above, the Examiner alleges that Morley discloses accessing the unified memory, a second time, with a data access during execution of the last instruction of the loop and points to Figure 28 to support the allegation.

With respect to Figure 28, the read operation illustrates that during the PHΦ of the MIO cycle, the data from the memory is accessed and place on the bus; during the PH1 of the MIO cycle, the data sits on the bus; and during the PH2 of the MIO cycle, the data on the bus is uploaded or read. Thus, the memory is only access once per MIO cycle and fails to anticipate accessing the unified memory, a second time, with a data access during execution of the last instruction of the loop, as set forth by independent claim 13.

In summary, with respect to independent claim 13, Morley fails to teach (1) accessing the unified memory with a dummy access during execution of the last instruction of the loop; and/or (2) accessing the unified memory, a second time, with a data access during execution of the last instruction of the loop.

Conclusion

Accordingly, for all the reasons set forth above, the Honorable Board is respectfully requested to reverse all the outstanding rejections. Also, an early indication of allowability is earnestly solicited.

Respectfully submitted,

Matthew E. Connors
Registration No. 33,298
Gauthier & Connors LLP
225 Franklin Street, Suite 2300
Boston, Massachusetts 02110
Telephone: (617) 426-9180
Extension 112

MEC/MJN/mjn

VIII. CLAIMS APPENDIX

1. A method for accessing a unified memory in a micro-processing system having a microprocessor, a one level pipeline, and a two-phase clock, such that an instruction is executed in a single instruction cycle, comprising:

(a) fetching a program instruction from the unified memory;

(b) determining if the fetched program instruction would require three unified memory accesses during a single instruction cycle for proper execution of the fetched program instruction, proper execution of the fetched program instruction being the microprocessor performing operations requested by the fetched program instruction in a single instruction cycle;

(c) accessing the unified memory a first time, during the instruction cycle associated with the fetched program instruction, with a dummy access when it is determined that the fetched program instruction requires three unified memory accesses for proper execution of the fetched program instruction;

(d) fetching a next program instruction from an instruction register, during the instruction cycle associated with the fetched program instruction from the unified memory, when it is determined that the fetched program instruction requires three unified memory accesses for proper execution of the fetched program instruction; and

(e) accessing the unified memory a second time, during the instruction cycle associated with the fetched program instruction from the first access of the unified memory, with a data access when it is determined that the fetched program instruction from the first access of the unified memory requires three unified memory accesses for proper execution of the fetched program instruction from the first access of the unified memory.

2. The method as claimed in claim 1, wherein the data access is a read data access.

3. The method as claimed in claim 1, wherein the data access is a write data access.

4. The method as claimed in claim 1, wherein the fetched program instruction from the first access of the unified memory is a last instruction of a loop.

5. The method as claimed in claim 1, wherein the instruction register is an instruction stack, thereby enabling program instruction fetches for nested loops.

6. A method for accessing a unified memory in a micro-processing system having a microprocessor, a one level pipeline, and a two-phase clock, such that an instruction is executed in a single instruction cycle, comprising:

(a) fetching a program instruction, corresponding to a second instruction cycle, from the unified memory during a first instruction cycle;

(b) determining if the fetched program instruction corresponding to the second instruction cycle is a conditional program code discontinuity;

(c) accessing the unified memory a first time during the second instruction cycle with a dummy access when it is determined that the fetched program instruction corresponding to the second instruction cycle is a conditional program code discontinuity; and

(d) accessing the unified memory a second time during the second instruction cycle to read a new instruction when it is determined the fetched program instruction corresponding to the second instruction cycle is a conditional program code discontinuity, thereby delaying instruction access from the unified memory for the second instruction cycle by a half cycle.

7. The method as claimed in claim 6, wherein the conditional program code discontinuity is a jump instruction.

8. The method as claimed in claim 6, wherein the conditional program code discontinuity is a call instruction.

9. A method for accessing a unified memory in a micro-processing system having a microprocessor, a one level pipeline, and a two-phase clock, such that an instruction is executed in a single instruction cycle, comprising:

(a) fetching a program instruction from the unified memory;

(b) determining if the fetched program instruction is a loop initiation instruction;

(c) storing a first instruction of the loop in an instruction register when the fetched program instruction is a loop initiation instruction;

(d) executing the loop;

(e) determining if a fetched instruction during the execution of the loop is a last instruction of the loop;

(f) accessing the unified memory a first time, during the instruction cycle associated with the fetched last instruction of the loop, with a dummy access;

(g) fetching the first instruction of the loop from the instruction register, during the instruction cycle associated with the fetched last instruction of the loop; and

(h) accessing the unified memory a second time, during the instruction cycle associated with the fetched last instruction of the loop, with a data access.

10. The method as claimed in claim 9, wherein the data access is a read data access.

11. The method as claimed in claim 9, wherein the data access is a write data access.

12. The method as claimed in claim 9, wherein the instruction register is an instruction stack, thereby enabling program instruction fetches for nested loops.

13. A method for accessing a unified memory in a micro-processing system during execution of a loop instruction, comprising:

(a) accessing a program instruction from the unified memory during a first instruction cycle;

(b) determining a type of program instruction;

(c) pre-fetching a second instruction from the unified memory;

(d) saving the pre-fetched instruction in a register when it is determined that the type of program instruction is a first instruction of a loop;

(e) fetching an instruction from the register when it is determined that the type of program instruction is a last instruction of a loop;

(f) accessing the unified memory with a dummy access during execution of the last instruction of the loop; and

(g) accessing the unified memory, a second time, with a data access during execution of the last instruction of the loop.

14. The method as claimed in claim 13, wherein the pre-fetched instruction is saved in a stack when it is determined that the type of program instruction is first instruction of a loop to enable nested loops and interruptible loops, and a next instruction is fetched from the stack when it is determined that the type of program instruction is a last instruction of the loop.

IX. EVIDENCE APPENDIX

NONE

X. RELATED PROCEEDINGS APPENDIX

NONE